Nonparametric Recognition of Nonrigid Motion

Ramprasad B. Polana and Randal C. Nelson

Technical Report 575
March 1995

UNIVERSITY OF
ROCHESTER
COMPUTER SCIENCE

Nonparametric Recognition of Nonrigid Motion

Ramprasad Polana and Randal Nelson

Department of Computer Science
University of Rochester
Rochester, New York 14627
Email: polana@cs.rochester.edu and nelson@cs.rochester.edu

Abstract

The recognition of nonrigid motion, particularly that arising from human movement (and by extension from the locomotory activity of animals) has typically made use of high-level parametric models representing the various body parts (legs, arms, trunk, head etc.) and their connections to each other. Such model-based recognition has been successful in some cases; however, the methods are often difficult to apply to real-world scenes, and are severely limited in their generalizability. The first problem arises from the difficulty of acquiring and tracking the requisite model parts, usually specific joints such as knees, elbows or ankles. This generally requires some prior high-level understanding and segmentation of the scene, or initialization by a human operator. The second problem, with generalization, is due to the fact that the human model is not much good for dogs or birds, and for each new type of motion, a new model must be hand-crafted. In this paper, we show that the recognition of human or animal locomotion, and, in fact, any repetitive activity can be done using low-level, non-parametric representations. Such an approach has the advantage that the same underlying representation is used for all examples, and no individual tailoring of models or prior scene understanding is required. We show in particular, that repetitive motion is such a strong cue, that the moving actor can be segmented, normalized spatially and temporally, and recognized by matching against a spatio-temporal template of motion features. We have implemented a real-time system that can recognize and classify repetitive motion activities in normal gray-scale image sequences. Results on a number of real-world sequences are described.

This work was supported in part by ONR (Office of Naval Research) under Grant N00014-93-I-0221 and in part by an ARPA subcontract from the University of Maryland (Z840902).

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	March 1995	technical report

4. TITLE AND SUBTITLE

Nonparametric Recognition of Nonrigid Motion

5. FUNDING NUMBERS

ONR N00014-93-I-0221

6. AUTHOR(S)

Ramprasad B. Polana and Randal C. Nelson

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESSES

Computer Science Dept.
734 Computer Studies Bldg.
University of Rochester
Rochester NY 14627-0226

8. PERFORMING ORGANIZATION

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESSES(ES)

Office of Naval Research
Information Systems
Arlington VA 22217

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

TR 575

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION / AVAILABILITY STATEMENT

Distribution of this document is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

(see title page)

14. SUBJECT TERMS

motion analysis; motion recognition; nonrigid motion; human movement

15. NUMBER OF PAGES

29 pages

16. PRICE CODE

free to sponsors; else $2.00

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
unclassified	unclassified	unclassified	UL

1 Introduction

Visual motion has long been considered a vital source of information in the computer vision literature. Understanding visual motion is important for distinguishing the sources of different motions, identifying objects moving relative to the surrounding environment, and also for navigational tasks such as obstacle detection and collision avoidance. Motion information is also useful for recovering the structure of the three-dimensional world and the three-dimensional motion of objects. Visual motion is a particularly effective cue for certain types of recognition tasks. This is due to the fact that algorithms for measuring characteristics of the motion field are relatively insensitive to illumination and shading.

The potential wealth of information derivable from visual motion has inspired a large body of work on the computation of exact geometric quantities such as the 3-D shape of objects, their location, and the motion of the observer. This reconstruction problem is sometimes referred to as the *structure-from-motion problem.* Solving the structure from motion problem implies recovery of the relative depth map and the three-dimensional motion parameters. This recovery is often more complex than the visual tasks that are supposed to utilize the information. Many nontrivial visual tasks however, do not require complete knowledge of the structure of the surrounding environment or the three-dimensional motion of the moving objects.

The emphasis on visual motion as a means of quantitative reconstruction of world geometry has tended to obscure the fact that motion can also be used for recognition. In fact, in biological systems the use of motion information for recognition is often more evident than its use in reconstruction. Humans have a remarkable ability to recognize different kinds of motion, both of discrete objects, such as animals or people, and in distributed patterns as in windblown leaves, or waves on a pond. The classic demonstration of pure motion recognition by humans is provided by Moving Light Display (MLD) experiments [17], where human subjects are able to distinguish activities such as walking, running or stair climbing, from lights attached to the joints of an actor. More subtle movement characteristics can be distinguished as well. For example, human observers can identify the actor's gender, and even identify the actor if known to them, by his or her gait.

Similar discrimination abilities using motion alone have been observed in non-human animals as well. A simple example occurs in the case of the common toad *Bufo bufo* , where any elongated object of a certain size exhibiting motion along the longitudinal axis, is identified as a potential food item, and elicits an orienting response [10]. Another example is the recognition of the female grayling butterflies by males. Tinbergen [32] reported that male butterflies fly towards crude paper models moving overhead, and that their response was not affected by the color or shape of the model. The key stimulus provoking males to fly towards the model turned out to be the pattern of movement: the flickering and up-and-down movements of a butterfly. Scout bees notify other bees of the direction of a feeding place they have discovered by means of a 'waggling dance', and the other bees recognize the dance and fly directly toward the food without deviating [34]. It has also been noted that bees approach oscillating flowers almost twice as fast as they do motionless ones [35]. Many organisms have elaborate courting ceremonies in which recognition of the type of movement or dance of the opposite sex, is crucial. Birds pay no attention to leaves and branches moving in the breeze, or during a storm, but they immediately observe the movement of a living person or animal in the midst of such an environment [30]. In general, a moving object is distinguished better than a motionless one.

Such abilities suggest that, in the case of machine vision, it might be possible to use motion as

a means of recognition directly, rather than indirectly through a geometric reconstruction. Model-based approaches have been proposed by earlier researchers to recognize visual motion, specifically biological ambulatory patterns, usually employing higher-level descriptions. The literature to date has not addressed the question of whether motion recognition can be accomplished using only low-level features of motion, and if so how it can be achieved and what features to use. We answer this question affirmatively, and provide specific demonstrations of motion recognition using such an approach. In an earlier paper, we demonstrated the utility of our approach in the context of *temporal textures* [25]. In this paper we concentrate on recognition of *activities* such as walking. Our goal is to show that motion recognition can be achieved using low-level robust features of motion without employing abstract models and without prior identification and tracking of representative feature points.

We have built a system that takes a discrete gray-level image sequence as input and determines whether any one of a known set of activities is present in the sequence. If so, the activity is identified. We hypothesize that there is an ideal motion pattern for each known class of activity. The ideal motion pattern corresponding to a specific class, undergoes various spatiotemporal transformations, combined with random variation, to give rise to the sample motion observed in an image sequence. Thus, we employ traditional statistical classification techniques using a feature vector based on visual motion that is normalized with respect to possible transformations. For instance, when the object producing the motion is not stationary, we utilize low-level motion information to track the object and keep it in the center of the image frame. The same information is also used to detect and compensate for changes in scale. We show that the selected features are robust and efficiently computable. In order to demonstrate the discrimination capabilities of the selected features, we use a simple nearest centroid classification.

A machine algorithm for recognizing such motions would be useful in applications, such as automated surveillance. Motion detection via image differencing can be used for intruder detection; however such systems are subject to false alarms, especially in outdoor environments, since the system is triggered by anything that moves, whether it is a person, a dog, or a tree blown by the wind. Motion recognition techniques can be used disambiguate such situations, and to build more reliable and accurate detection systems. Another application is in industrial monitoring. Many manufacturing operations involve a long sequence of simple operations, each performed repeatedly, at high speed, by a specialized mechanism at a particular location. It should be possible to set up one or more fixed cameras that cover the area of interest, and to characterize the permitted motions in each region of the image(s). Such automated systems can be used in hazardous plant conditions and can also be a cheap alternative compared to manual systems. They would also detect critical breakdowns immediately rather than after a number of defective units have been produced. Lip-reading systems based on motion recognition can be used for improving the recognition rate of utterances, by characterizing motion of the lips and motion surrounding the mouth area for different utterances. For instance, employing the feature vectors computed by the algorithms described here alone, Virginia de Sa [7] obtained a recognition rate of approximately 70 percent in this application. Recognition of hand gestures can aid in designing better man-machine interfaces. Rhyne and Wolf [28] study the use of gestures for editing operations and menu-oriented operations of pointing, selecting etc. On-line handwritten character recognition can be made easier by recognizing the motion trajectories of the hand for different characters.

The following section discusses the related literature concerning motion recognition. In section 3, we demonstrate low-level recognition of activities while the object of interest remains stationary

in the image frame. In section 4, we describe tracking and normalization with respect to scale changes for non-stationary objects. We describe the complete algorithm in section 5. Section 6 presents conclusions and discusses some extensions and possible future research directions.

2 Related Work

Though motion plays an important role in recognition tasks, motion recognition in general, has received relatively little attention in the literature. Most computational motion work, as mentioned previously, has been concerned with various aspects of the structure-from-motion problem. There is a large body of psychophysical literature addressing the perception of motion, most of it concerned with primitive percepts. A modest amount of this work addresses more complicated motion recognition issues [17; 6; 16; 15], but the models and descriptions have typically not been implemented. Various computational models of temporal structure, have been proposed (e.g. [4; 11]) but much of this work is at a fairly high level of abstraction, and has not actually been applied to visual motion recognition except in rather artificial tests.

A few researchers have worked towards obtaining higher-level descriptions, usually employing a linguistic approach for representing motion concepts [3], [20]. Badler provides a computational methodology for the description of events based on object movements. Trajectories of idealized point features through successive image frames are associated with different motion verbs such as dropping, throwing, approaching etc. Badler is not concerned with low-level image processing, instead he assumes that the point-like object features are supplied with each frame. Koller et al. describe motion of vehicles in traffic scenes using motion verbs such as start, brake, brake without stopping, turn, move away etc. The object behavior as given by its observed trajectory is associated with corresponding motion verbs. The trajectories are obtained by tracking features, which are centroids of blobs generated by a monotonicity operator in the image frame.

Human motion, specifically walking, has been studied extensively using model-based approaches [22], [16]. O' Rourke and Badler show how human motion can be tracked using a detailed model of the human figure and a top-down approach for prediction and constraint propagation. Low-level image analysis is limited to searching specified image areas for certain body features employing various feature detectors or locators. Hoffman and Flinchbaugh propose a computational approach to understand biological ambulatory patterns utilizing specific anatomical constraints. The limbs of animals, for instance, typically move in single planes for extended periods of time.

Motion recognition has been studied in the context of MLDs (moving light displays), with the goal of automating the capabilities of the human visual system. The demonstrations using MLDs show that trajectories of a few specific points corresponding to joints of an actor performing an activity, can provide a key for recognizing the activity. Goddard addresses recognizing the activity in MLDs involving single actors moving parallel to the image plane using a connectionist approach. His work addresses the representation of motion event sequences and their recognition assuming certain invariant image features. His input consists of the joint angles and angular velocities computed from the motion of the dots in the light displays. The joint angles and angular velocities are invariant to rotation in the image plane, scale and translation. A challenging part in computing these invariants is to recover the connectivity of the individual dots (by body parts) in the MLD images. A domain independent approach to this problem is given by Rashid. Rashid [27; 22] considered the computational interpretation of moving light displays, particularly in the context of gait determination. This work emphasized rather high-level symbolic models of temporal sequences,

an approach made possible by the discrete nature of the moving light displays. The results were quite sensitive to discrete errors and thus highly dependent on the ability to solve the correspondence problem and accurately track joint and limb positions. This severely limits the general applicability of the method.

Trajectories of specific points belonging to an object have been used for motion recognition in contexts other than MLDs as well. Gould and Shah [14] represent motion characteristics of moving objects by recording the important events in their trajectory. The use of the resulting *trajectory primal sketch* in a motion recognition system is demonstrated by Gould et al. in [13]. The curvature features of trajectories have been used to detect cyclic motion by Allman and Dyer [1] and by Tsai et al. [33]. Recently, Seitz and Dyer [29] describe an algorithm for detecting periodic motion under arbitrary affine transformations of the object.

Koller, Heinze and Nagel [20] developed a system that tracks moving vehicles and characterizes their trajectory segments in terms of natural language concepts. The trajectories are obtained by tracking features, which are centroids of blobs generated by the monotonicity operator.

Very few researchers have addressed motion recognition directly using purely low-level features of retinal motion information. Anderson et al. [2] describe a method of change detection for surveillance applications based on the spectral energy in a temporal difference image. This has the flavor of the temporal texture analysis described here, but was not generalized to other motion features or more sophisticated recognition.

A few studies have considered highly specific aspects of motion recognition computationally. Some temporal pattern recognition work has been done in the context of speech processing [18; 31; 9], but the applicability of the techniques to motion recognition has not been considered. However, a few studies in speech recognition using visual cues (lip-reading) are relevant to motion recognition. Petajan et al. [24] use images of mouth opening over time while Finn and Montgomery [12] and Mase and Pentland [23] track points around the mouth and use features of specific points to characterize the spoken words. The system described by Pentland recognizes spoken digits with 70%-90% accuracy over 5 speakers. The system required the location of the lips to be entered by hand, and depended on an explicitly constructed lip model.

3 Recognizing Stationary Activities

3.1 Activities

Activities involve a regularly repeating sequence of motion events. If we consider an image sequence as a spatiotemporal solid with two spatial dimensions x, y and one time dimension t, then repeated activity tends to give rise to periodic or semi-periodic gray level signals along smooth curves in the image solid. We refer to these curves as *reference curves*. If these curves could be identified and samples extracted along them over several cycles, then frequency domain techniques could be used in order to judge the degree of periodicity. (Technically the above is true only in conditions where scaled orthographic projection is a good approximation to the viewing conditions associated with the object of interest, but this is valid for most practical situations involving the observation of moving objects.)

Before defining the reference curves, we formalize the concept of a periodic object. An object is defined as a set of points P. For purposes of the current discussion, this set can be considered to represent the surface of the object, since these are the points that can contribute to the object's

appearance. Associated with each $p \in P$ is a function $X_p(t)$ giving its location (in a fixed 3D coordinate system) as a function of time. A stationary periodic object (ie. a stationary object exhibiting periodic activity) has the property that $X_p(t) = X_p(t+\tau)$ for all $p \in P$, where τ is the time period for one cycle of the activity and is independent of p. We now define a translating periodic object. Such an object has the property that $X_p(t) = Y_p(t) + Z(t)$, where Y_p satisfies $Y_p(t) = Y_p(t+\tau)$ and $Z(t)$ is a path in 3D space independent of p. It can be assumed that $Z(0) = \mathbf{0}$ so that $X_p(0) = Y_p(0)$. Intuitively, a periodic object characterized by $Y_p(t)$ is translated along the path $Z(t)$. If we compensate for the translation of the object, we would be looking at a stationary periodic object described by the equation: $X_p(t) - Z(t) = Y_p(t) = Y_p(t+\tau) = X_p(t+\tau) - Z(t+\tau)$. Note that $Z(t)$ is not necessarily periodic. Note also that a stationary periodic object is a special case of translating periodic object with no translation, or in other words $Z(t) = \mathbf{0}$ for all t.

Now suppose the viewing situation can be well described by scaled orthographic projection, which means, essentially, that the ratio of the object distance to the object size (depth) remains large during the period of observation. In this case, we can associate a plane W parallel to the image plane that contains the object at $t = 0$. For each point w in this plane, we can define a 3D-reference curve $R_w(t)$ to be the path $w + Z(t)$. We define the 2D-reference curve $r_w(t)$ corresponding to the same point to be the projection of $R_w(t)$ onto the image plane over time (hence $r_p(t)$ is a curve in (x, y, t) space).

Under scaled orthographic projection of the object, the gray-level signal on the 2D-reference curve $r_w(t)$ at any time t is determined by the set of object points $q \in P$ such that $X_q(t)$ projects to $R_w(t)$, where the direction of projection is perpendicular to the image plane. In particular, if the set is non-empty, the gray level observed is determined by the point in the set closest to the observer; if the set is empty, the gray level is determined by the background. All this is just a formal way of saying that the object may self-occlude in a time-varying way, and the observer sees the closest point along a given projection.

It can be shown that, for any reference curve in the image, any points of the object that occur along it (in the above sense) must do so periodically. This means that the gray-level signal along any reference curve that contains only points from the object will be periodic with period τ. If the reference curve only sometimes contains points from the object, then the gray-level signal will consist of a periodic part due to the object added to a (non-correlated) non-periodic part due to the background. We term this a semi-periodic signal.

By way of illustration, consider a walking person. This is an example of a non-stationary activity; that is, if we attach a reference point to the person, that point does not remain at one location in the image. If the person is walking with constant velocity, however, and is not too close to the camera, then the reference point moves across the image on a path composed of a constant velocity component modulated by whatever periodic motion the reference point undergoes. Thus, if we know the average velocity of the person over several cycles, we can compute the spatiotemporal line of motion along which the periodicity can be observed. If the person moves with average velocity (u, v) the spatiotemporal line of motion will be determined by the equations $(x, y) = (u, v) * t + (x_0, y_0)$, where (x, y) is the position of the object in space at time t and (x_0, t_0) is the position at time zero. This applies to any object undergoing constant velocity locomotion.

3.2 Periodicity Detection

A periodic motion detection algorithm under arbitrary affine transformations of the object was reported in [29]. In this section, we briefly describe the periodic activity detection algorithm used in our experiments and reported in [26]. We use Fourier methods to determine periodicity in an image sequence. Non-stationary objects are tracked and then the periodicity measure for the activity is computed.

From Fourier theory we know that any periodic signal can be decomposed into a fundamental and harmonics. That is, we can consider the energy of a periodic signal to be concentrated at frequencies which are integral multiples of some fundamental frequency. This implies that if we compute the discrete Fourier transform of a sampled periodic signal, we will observe peaks at the fundamental frequency and its harmonics. Hence, in theory, the periodicity of a signal can be detected by obtaining its Fourier transform and checking whether all the energy in the spectrum is contained in a fundamental frequency and its integral multiples.

The real-world signals, however are seldom perfectly periodic. In the case of signals arising from activity in image sequences, disturbances can arise from errors in the uniform translation assumption, varying background and lighting behind a locomoting actor, and other sources. In addition, for computational purposes, we need to truncate the signal at some finite length which may not be an exact integral multiple of its period. Nevertheless, the frequency defined by the highest amplitude often represents the fundamental frequency of the signal. Hence we can get an idea of the periodicity in a signal by summing the energy at the highest amplitude frequency and its multiples, and comparing that quantity to the energy at the remaining frequencies. Such a measure is defined for each signal and a method of computing periodicity in an image sequence is described in [26]. The measure is normalized with respect to the total energy at the frequencies of interest so that it is one for a completely periodic signal and zero for a flat spectrum. The periodicity for an image sequence is obtained by combining the periodicity measures of motion signals extracted along the reference curves. For completeness, we list the steps of the algorithm for deciding if any periodic activity exists in an image sequence are given below.

- *Input:* The input to the algorithm is a digitized image sequence consisting of 128 frames of resolution 128x128 pixels.
- *Output:* A periodicity measure indicating the amount of periodicity in observed in the image sequence. This is used to decide whether the image sequence contains a periodic activity and if so, to locate the region of the activity.
- *Step 1.* Compute normal flow magnitude at each pixel between each successive pair of frames using the differential method.
- *Step 2.* Mark pixels corresponding to significant motion in the scene by thresholding the normal flow magnitude. Compute the centroid of the marked pixels in each frame. Compute the mean velocity (if any) of the actor by fitting a linear trajectory to the sequence of centroids. Take the reference curves to be the lines in the spatiotemporal solid parallel to this linear trajectory.
- *Step 3.* Extract motion signals along the reference curves. Compute the dominant frequency w and the periodicity measure P_w for each individual signal extracted.

- *Step 4.* Compute overall periodicity measure P for the image sequence using the formula described above.

The effectiveness of the algorithm was demonstrated through experiments using real-world image sequences in [26]. The periodicity measures computed using the above algorithm are plotted for 20 periodic and all 8 non-periodic sequences in Figure 1. As is evident from the graphs and the projected scatter plot, the technique separates complex periodic from non-periodic motion cleanly. The requirement that an empirically determined threshold be used is not a great drawback in this case, nor is it particularly surprising, since even the the intuitive notion of periodic activity falls on a continuum. Is the motion of a branch waving somewhat irregularly in the wind periodic or non-periodic? Here, we classified it as non-periodic, but it had one of the higher periodicity measures, as might be expected.

In our experiments, the periodic activity samples consist of at least four cycles of the activity. A minimum of four cycles were used to detect the actual frequency given that there is a considerable amount element of non-repetitive structure from the background in the case of translating actors. We assumed that any activity that existed in the data would be either stationary, or locomotory in a manner that produced an overall translating motion. We also assumed that there was at most one actor in the scene, though a certain amount of background motion could be tolerated. A third assumption is that the viewing angle and the scene illumination does not change significantly so that the intensity along the reference curves remains periodic.

3.3 Recognition

In this section, we describe a robust method for recognizing stationary activities. We shall extend the technique described here to non-stationary activities by means of normalization with respect to spatial translation and scale. For this section, we shall assume that the input image sequence contains a stationary activity and proceed to describe the recognition scheme. The recognition scheme is based on low-level features of motion, and does not require the recognition or tracking of specific parts of the actor. We make use of the motion field computed between successive frames as a basis to extract a feature vector and use it for classification of the activity.

The periodicity detection procedure provides a periodicity measure for each active pixel in a tracked object. By back-projecting this measure, we can locate the pixels in each frame that display periodicity at the dominant frequency. Since these pixels are likely to belong to the actor of interest, we can use this back-projection to refine our initial segmentation, which was based solely on aggregate motion. The fundamental frequency allows us to frame the activity in time, and compensate for variation in temporal scale (i.e. frequency). The result of such normalization is a spatiotemporal solid containing the activity of interest in a form that is invariant to temporal scale. The next step is to compute a descriptor for this solid that can be used to classify the activity it represents. Such a descriptor should capitalize on the fact that a periodic activity is characterized by regularly repeating motion events that have fixed spatial and temporal relationships to each other.

We divide one cycle of the spatiotemporal solid into XxYxT cells by partitioning the two spatial dimensions into X and Y divisions respectively, and the temporal dimension into T divisions. In the experiments described below, we used four divisions in each spatial dimension, and six divisions along the temporal dimension, resulting in a feature vector of length 96. We then select a local motion statistic and compute the same statistic in each cell of the spatiotemporal grid. The feature

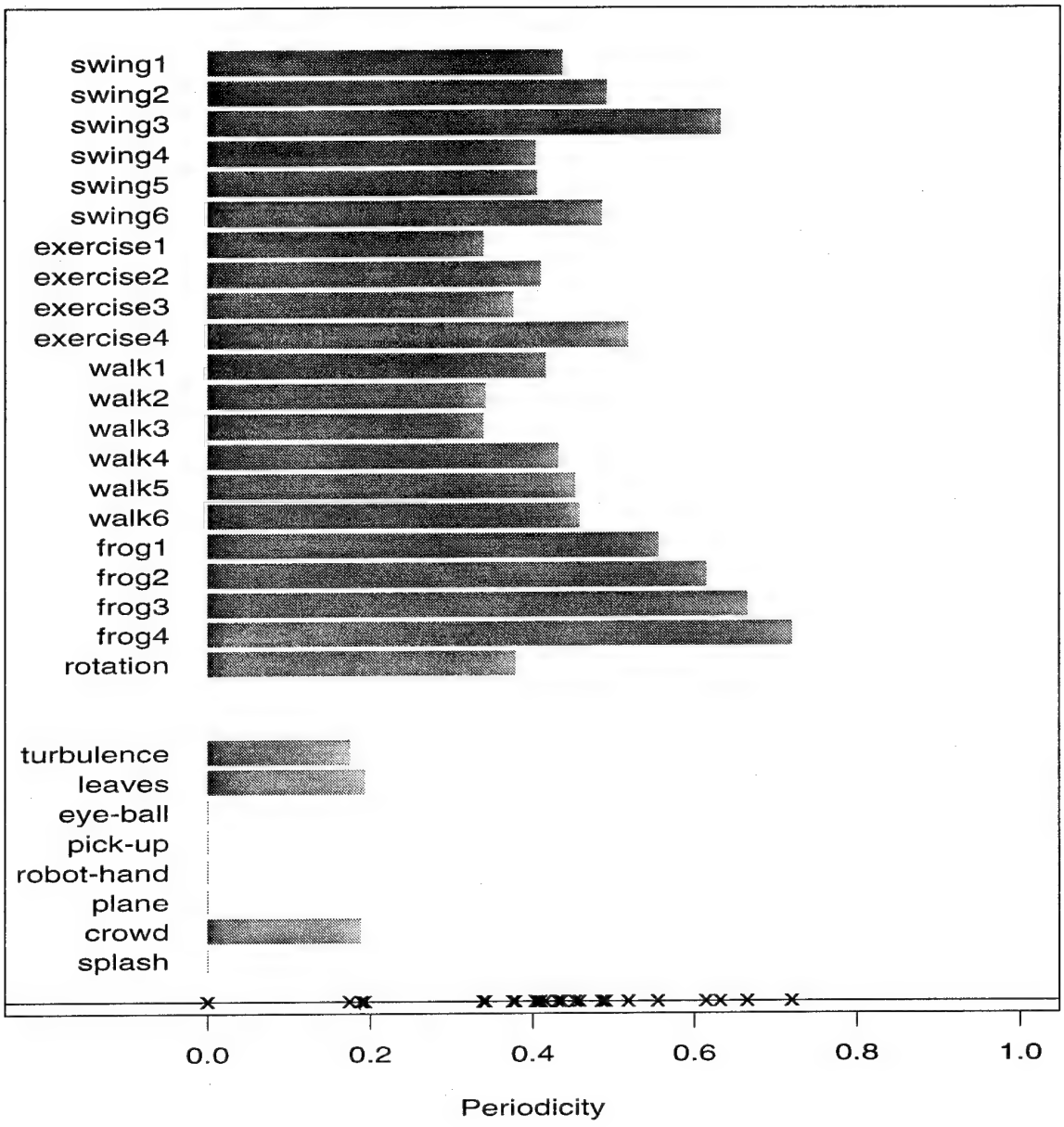

Figure 1: Periodicity measure for Periodic and Nonperiodic sequences

vector in this case is composed of XYT elements each of which is the value of the statistic in a particular cell – essentially a three dimensional template.

One issue that affects the measures described above is the fact that the normalized spatiotemporal solid, while corrected for temporal scale (frequency) is not corrected for temporal translation (phase). There are a couple of ways to handle this. One is to pick some robust temporal feature to define zero phase, and normalize all samples with respect to this feature. One feature that works fairly robustly is to take the time of maximum difference between total motion in the left and right half fields. Alternatively, since the pattern matching phase of the algorithm currently represents only a small fraction of the total computational effort, and the temporal resolution of the pattern is typically small (i.e., less than 10 samples per cycle), we can simply try a match at each possible phase and pick the best. We have found in our experiments that this method works better than the first. The results that follow use this kind of matching.

We experimented with three different local statistics. The first statistic is simply the summed normal flow magnitude in each cell. The directional information is ignored in this case. The second statistic is the dominant motion direction in each cell. This is approximated by computing the histogram of normal flow directions weighted by the corresponding normal flow magnitude and selecting the direction with highest histogram value. The third statistic represented the summed motion magnitude in the dominant motion direction.

3.4 Experimental Results

We ran experiments on seven different types of activities. The image sequences were first recorded on video and then digitized later with suitable temporal sampling so that at least four cycles of the activity were captured in 128 frames. Following is a description of each activity and the conditions under which they were digitized.

- Walk: A person walking on a treadmill.
- Exercise: A person exercising on a machine.
- Jump: A person performing jumping jacks.
- Swing: A person swinging viewed from the side.
- Run: A person running on a treadmill.
- Ski: A person skiing on a skiing machine.
- Frog: A toy frog simulating swimming activity viewed from above.

All samples were digitized at a spatial resolution of 128x128 pixels, except those for walk and run which were digitized at a resolution of 64x128 pixels. Pixels were 8 bit gray levels. The swing and exercise activities were shot outdoors and contained background motion. Image frames from a sample each of the seven activities are illustrated in Figure 2.

We first digitized eight samples of each activity by the same person under the same conditions with respect to scene illumination, background, and camera position. We created the reference database taking half of the samples belonging to each activity. In other words, the reference database consists of four samples of each of the seven activities. The remaining four samples of

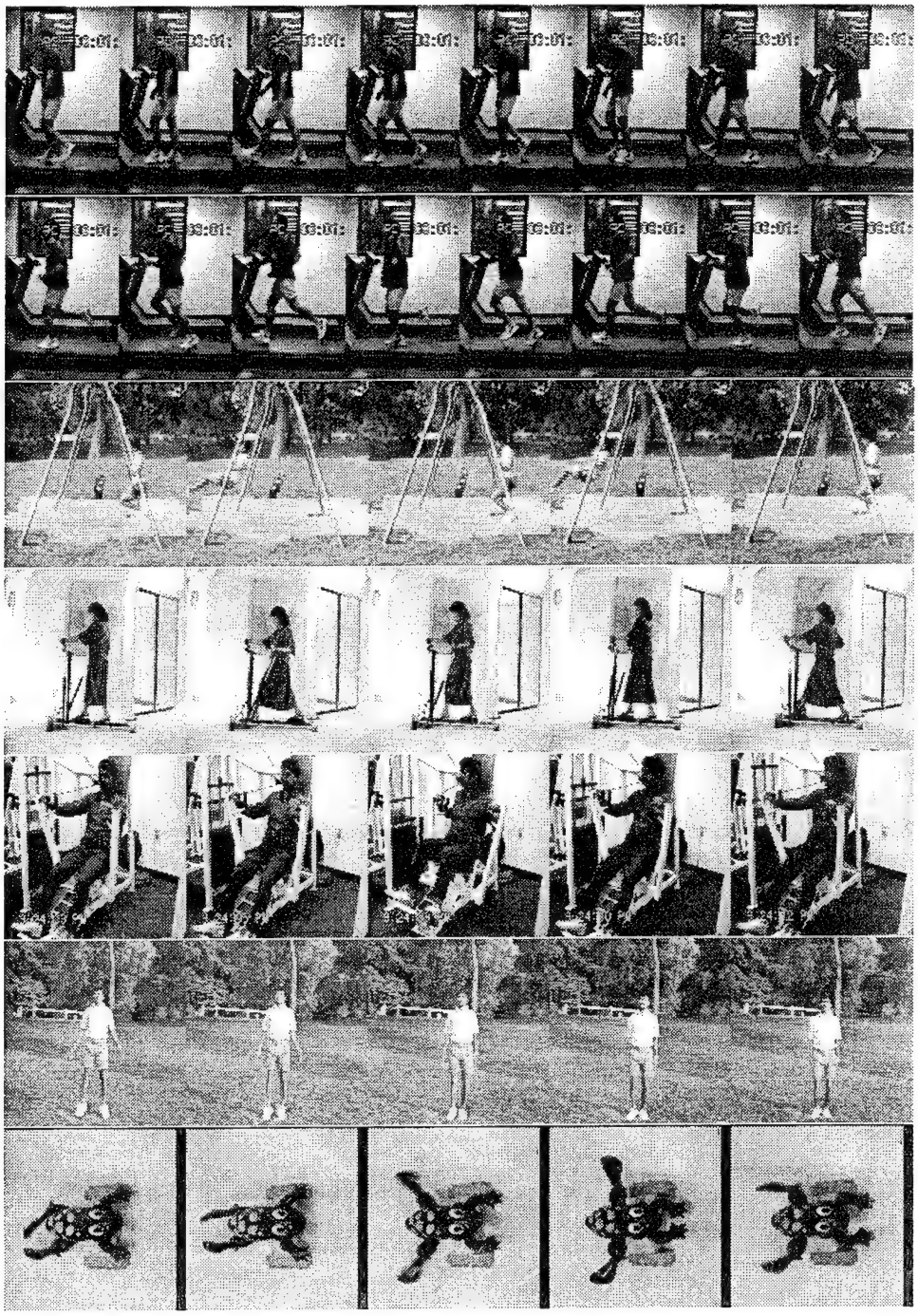

Figure 2: Sample images from periodic activities: walk, run, swing, jump, ski, exercise and toy frog

Figure 3: Sample images of walk by a different actor and toy frog with different background and frequency

each activity are used to create the test database. In addition, we digitized four samples of walking by a different person and eight samples of the frog under different lighting conditions and different background and foreground patterns. These samples also differed from the reference database in frequency, speed of motion, and spatial scale. Examples of these samples are shown in Figure 3 These samples were added to the test database. The samples in the test database were classified by a nearest centroid classification technique using the samples in the reference database as training set.

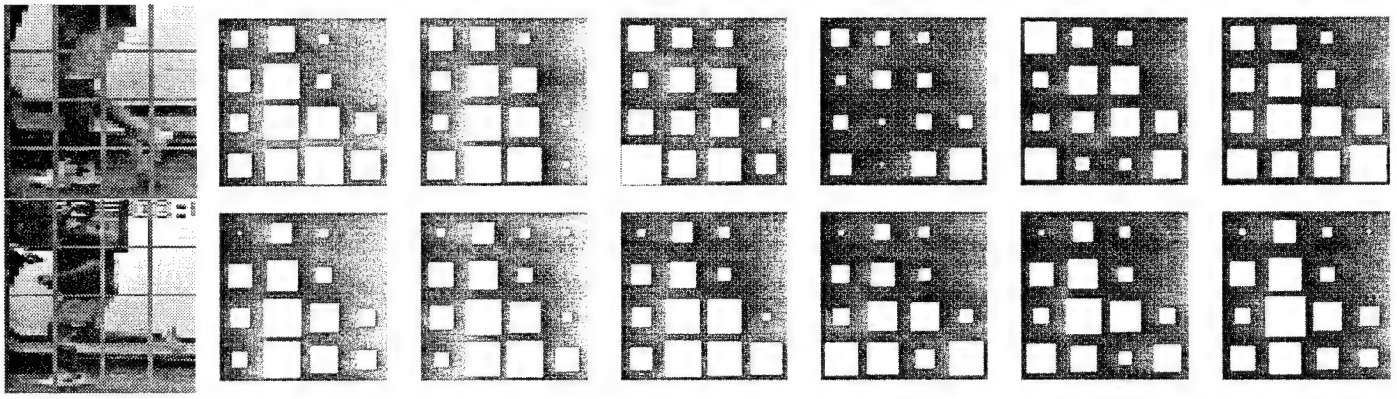

Figure 4: Total motion magnitude feature vector for a sample of walk (top) and a sample of run (bottom). One cycle of activity is divided into six time divisions shown horizontally. Each frame shows the spatial distribution of motion in a 4x4 spatial grid (size of each square is proportional to the amount of motion in the neighborhood).

We conducted experiments with three feature vectors, described above. Sample features vectors are illustrated in Figure 4 using the total motion magnitude statistic for a walk and a run sequence. The results of classification using different variations are shown in terms of percentage of test cases correctly classified in Table 1. The percentage of correct classification does not give a good indication for the quality of classification. Hence, we also illustrate the results by the confusion matrix which shows how closely test samples belonging to various classes match the reference

samples of those classes. The confusion matrix using the feature vector of total motion magnitudes is shown in Figure 5.

The best results were obtained using the summed motion magnitudes. This might seem surprising, given that there is presumeably more information available when directional information is used as well. Further investigation revealed that the resolution of our original sequences was not great enough to allow directional information to be reliably computed. In other words, for any given motion magnitude and direction, there were so few pixels contributing that the small sample size coupled with the high intrinsic error in differentially computed flow eliminated any advantage lying in the higher dimensionality. With much higher resolution pictures, this problem is reduced, but a lot more processing is required. The classification based on the summed magnitudes resulted in correct classification of every sample in the test database, including the samples using a different actor and different backgrounds, which were not represented in the reference database i.e., 100% correct.

Feature vector	*Total Test Samples*	*Correct Classified*	*Percent Success*	*Failures*
maximal motion direction	40	32	80	walk by different actor and frog with different patterns
maximal motion magnitude	40	39	97.5	walk by different actor
total motion magnitude	40	40	100	None

Table 1: Classification results

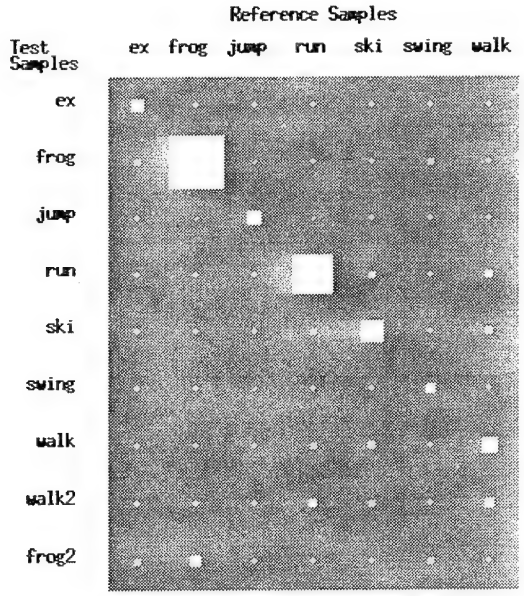

Figure 5: Confusion matrix for the feature vector using total motion magnitude

Added Clutter Percentage	*Total Test Samples*	*Successfully Detected*	*Correctly Classified*
25	4	3	3
50	4	3	3
75	4	2	0
100	4	2	0
150	4	1	0
200	4	0	0

Table 2: Classification results with motion clutter (samples are of walk)

3.5 Clutter Tolerance

A classification success rate of 100% does not reveal much about the robustness of the algorithm. To understand how much background clutter can be tolerated by this technique, we experimented with the walk samples by adding motion clutter produced by leaves fluttering in the wind. We feel that this sort of interference is more typical of the situations that would affect motion recognition in applications than (say) Gaussian noise. The normal motion field computed from these leaf sequences was added, vectorwise, to that computed for the test sequences in a controlled fashion so that its mean magnitude represented a varying percentage of the mean magnitude of the signal. The resulting samples were then classified using a feature vector of total motion magnitudes, as described earlier. The mean magnitude of the motion clutter was increased until the recognition algorithm could no longer classify the activity correctly. The results are tabulated in Table 2. The results show that the recognition scheme can tolerate motion clutter whose magnitude is equal to one half that of the activity, and it displays degraded, but still useful performance for even higher clutter magnitudes.

4 Spatial and Temporal Normalization

We used a spatiotemporal motion magnitude template as a basis for the recognition of activities. In order for this to work, the motion to be identified must be normalized with respect to spatial scale, spatial translation and temporal scale and translation. Template matching is a well studied and frequently effective method of recognition. It fails when sufficiently rigid normalization cannot be carried out. It turns out that periodicity inherent in motion such as walking or running is a sufficiently strong cue to allow strong normalization to be performed.

Given a gray valued image sequence, we first detect pixels corresponding to independently moving objects. These pixels are grouped using spatial clustering methods, and an object of interest (the actor performing the activity) is selected and tracked. The subsequent activity detection and recognition can be applied to each independently moving object. For each selected object, a spatiotemporal template of motion features is obtained from the motion of the corresponding pixels, and it is used to match the test sample with the reference motion templates of known activities.

Spatial translation invariance is achieved by tracking the selected object through successive frames, estimating the spatial translation parameters and compensating for the translation of the object. The spatial translation parameters are estimated by assuming the object is moving along locally a linear trajectory and using a least squares technique. Spatial scale invariance is achieved

by measuring the spatial size of the object through successive frames, estimating the spatial scale parameters and compensating for the changes in scale. This method is described in detail in the next subsection.

Temporal scale invariance is achieved by detecting the frequency of the activity and obtaining one cycle of activity by averaging motion information of multiple cycles. Temporal translation has turned out to be hard to estimate from the motion information, but it was handled in the matching stage by matching the test template with reference template at all possible temporal translations.

The following sections describe the steps of the algorithm in greater detail.

4.1 Detecting Independently Moving Objects

In order to apply the motion recognition algorithm, the moving actor must be segmented out of the scene. If there are multiple actors in the scene, the recognition algorithm should be applied to each of them separately. If the motions do not interfere with each other, they can be segmented and tracked separately. By using a predictive tracker, an occasional crossing of different activities can be tolerated as long as the regions can be separated again later. For this, it is important to initially detect each actor in the scene and spatially isolate them.

Fortunately, independent motion provides an exceptionally strong segmentation cue. Nelson [21] has demonstrated a real-time method of detecting independently moving objects even in the case that the observer is itself moving. Using such a method, we can detect the pixels in an image sequence that exhibit motion independent of that of the background and segment the image frames into distinct regions corresponding to different moving objects. Some other common methods of segmenting multiple moving objects are: using color cues, distance from camera obtained from a range sensor, or selecting objects moving with a certain velocity (motion measurements can be clustered to produce image segmentation and the segment that conforms to the desired range of motion selected). The particular selection method is not important and as long as there is a method of selecting an actor of interest, we can restrict our attention to tracking a single actor using the algorithm given below.

4.2 Tracking in the Presence of Other Moving Objects

If there is a single moving object in the scene, we can track it by making use of the centroid of the moving pixels. Using the whatever motion detection scheme is appropriate for the situation, we compute the centroid of the motion

$$(x_t, y_t) = \sum_{(i,j,t)\in S(t)} (i, j)/N$$

where $S(t)$ is the set of pixels in frame t that correspond to the moving object and N is the number of pixels in $S(t)$. The centroid measurements can be used to estimate the translation parameters of the object and obtain a locally linear smooth trajectory using the model $(x_t, y_t) = (x_0, y_0)+(u, v)*t$, where (u, v) is the local velocity of the object.

Such a simple method of tracking may not work if there is more than one moving object in the field of view. If we only use the centroids of motions to track, the centroid could follow another object that temporarily occludes the object of interest. To avoid such a situation, we make use of an estimate of shape of the object and its predicted position in the image frame to restrict the centroid computation to the area that is most likely corresponds to the object.

The algorithm is described below: Suppose $S(t)$ is the set of pixels that corresponds to the estimated object, and (x_t, y_t) is the position of the object in flow frame t. From the position estimates of the past K flow frames, we obtain an estimate of the velocity of the object (assuming local linear translatory motion as before). Let (u_t, v_t) be the velocity estimate at flow frame t. Then the predicted position of the object in flow frame $t+1$ is

$$p(t+1) = (x_t + u, y_t + v).$$

An estimate for the set of pixels corresponding to the object in flow frame $t+1$ is

$$S'(t+1) = \{(x+u, y+v) : (x,y) \in S(t)\}.$$

We measure the centroid of motion

$$c(t+1) = \sum_{(i,j,t)\in S'(t+1)} (i,j)/\|S(t)\|$$

in frame $(t+1)$, and then estimate the updated position of the object in flow frame $t+1$ to be

$$(x_{t+1}, y_{t+1}) = w * p(t+1) + (1-w) * c(t+1)$$

where w is between 0 and 1. This is repeated for every frame, estimating the position of the object using its estimated velocity from past K frames and centroid of motions in the current frame.

A demonstration of the tracking algorithm in the presence of multiple moving objects and occlusions by other objects is shown in Figure 6. The illustration shows an image sequence consisting of two persons walking towards and crossing each other. The object of interest in this case is the person walking from right to left. The first eight frames of the 64 frame image sequence contain only one person, and this section is used to bootstrap the tracker. Later, a second person temporarily occludes the first. In the first eight frames, we thresholded the motion to highlight significant motion. Using those locations we estimated the shape of the object of interest in the form of a rectangle surrounding the object, which is illustrated in the figure. Successive estimated positions are shown with a plus (+) sign. It can be seen that the tracking algorithm smoothly tracks the first person even when there is occlusion. Figure 7 shows the tracking during the occlusion.

4.3 Eliminating Background Motion

Estimating the trajectory of an object and compensating for its translation so that the object remains in the center of the image causes the previously stationary background to appear to be moving. In the compensated image frames, the background will be moving with the same velocity magnitude as the object velocity estimated but in the opposite direction. After computing the flow fields between successive frames of the compensated image sequence, we eliminate any motion that is consistent with the background velocity by making the estimate at that point unknown.

Alternatively, instead of translating the image frames and recomputing the flow frames, we could update the flow field by subtracting the estimated trajectory motion (u, v) of the object. Such a subtraction, however leads to large inaccuracies in the measured flow, primarily because the differential techniques we use for speed have low accuracy, and subtraction can cause the values to lose all significance. Hence, we chose to recompute the motion measurements after translating the image frames.

Figure 6: Tracking in the presence of multiple moving objects.

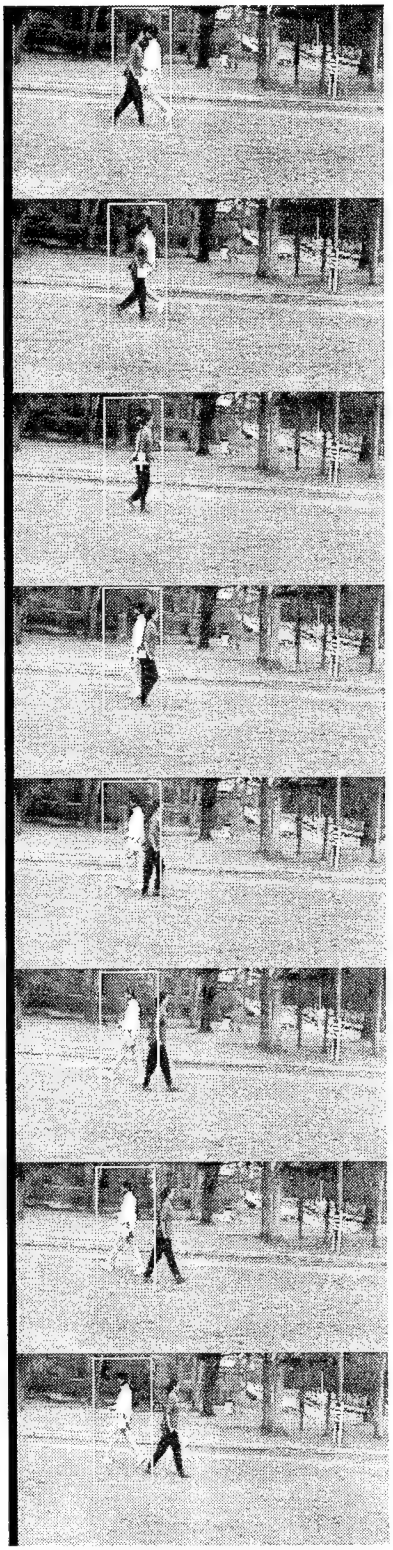

Figure 7: Detail of tracking in the presence of temporary occlusion.

4.4 Changing Scale

In this section, we show how the changes in spatial scale of the activity can be detected and compensated for. We make a key assumption here: that the height of the object of interest does not change through the image sequence. This is certainly true for the activities of human walking, running etc., and it is a reasonable assumption for a host of other activities. (Even when the height is changing, the periodic repetition of the activity requires that the same height recur through successive cycles of the activity, and hence fitting the model described below over many periods in this case will give good estimates of scale changes).

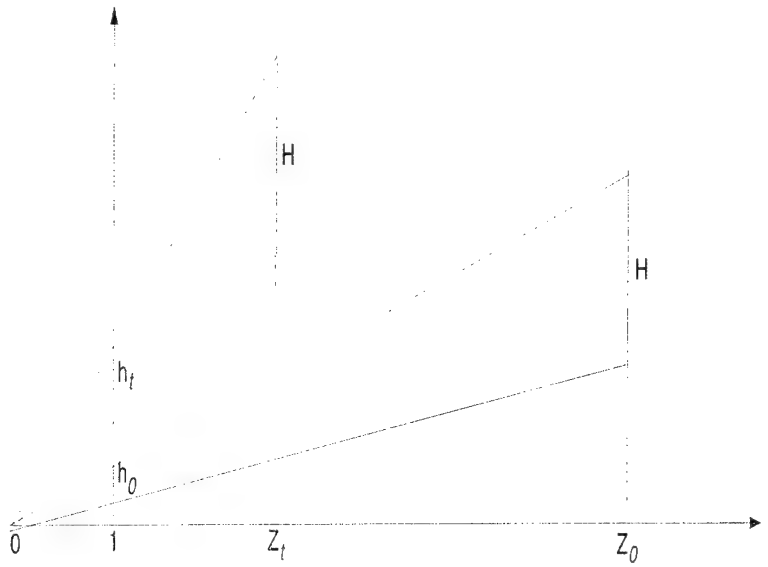

Figure 8: Model of changing scale based on object height

The model of projected image height of the object is illustrated in Figure 8. Let H be the actual height of the object in three-dimensional world (assumed not to change). According to the imaging model illustrated in the figure, the image coordinates (x_t, y_t) in image frame t are related to the three-dimensional world coordinates (X_t, Y_t, Z_t) by $(x_t, y_t) = (X_t/Z_t, Y_t/Z_t)$, where Z_t is the distance of the object from the camera at image frame t. From this we can derive that $h_t = H/Z_t$ where h_t is the projected image height of the object at image frame t. Now, if we assume the object is approaching or moving away from camera at a locally constant velocity, say W, then $Z_t = Z_0 + W * t$ is the distance of the object from camera. Using the relation $h_0 = H/Z_0$, we find the image height of the object over time to be

$$h_t = h_0/(1 + w * t)$$

where $w = W/Z_0$ is a constant scale factor. (Note that w is negative if the object is approaching the camera, positive if the object is moving away, and it is exactly equal to zero if the object's distance from the camera does not change).

By estimating height of the object in each flow frame and using the model described above, we obtain an estimate for the locally constant scale factor w and then compensate for the scale changes by scaling the image frame t so as to match the scale of the activity in the reference

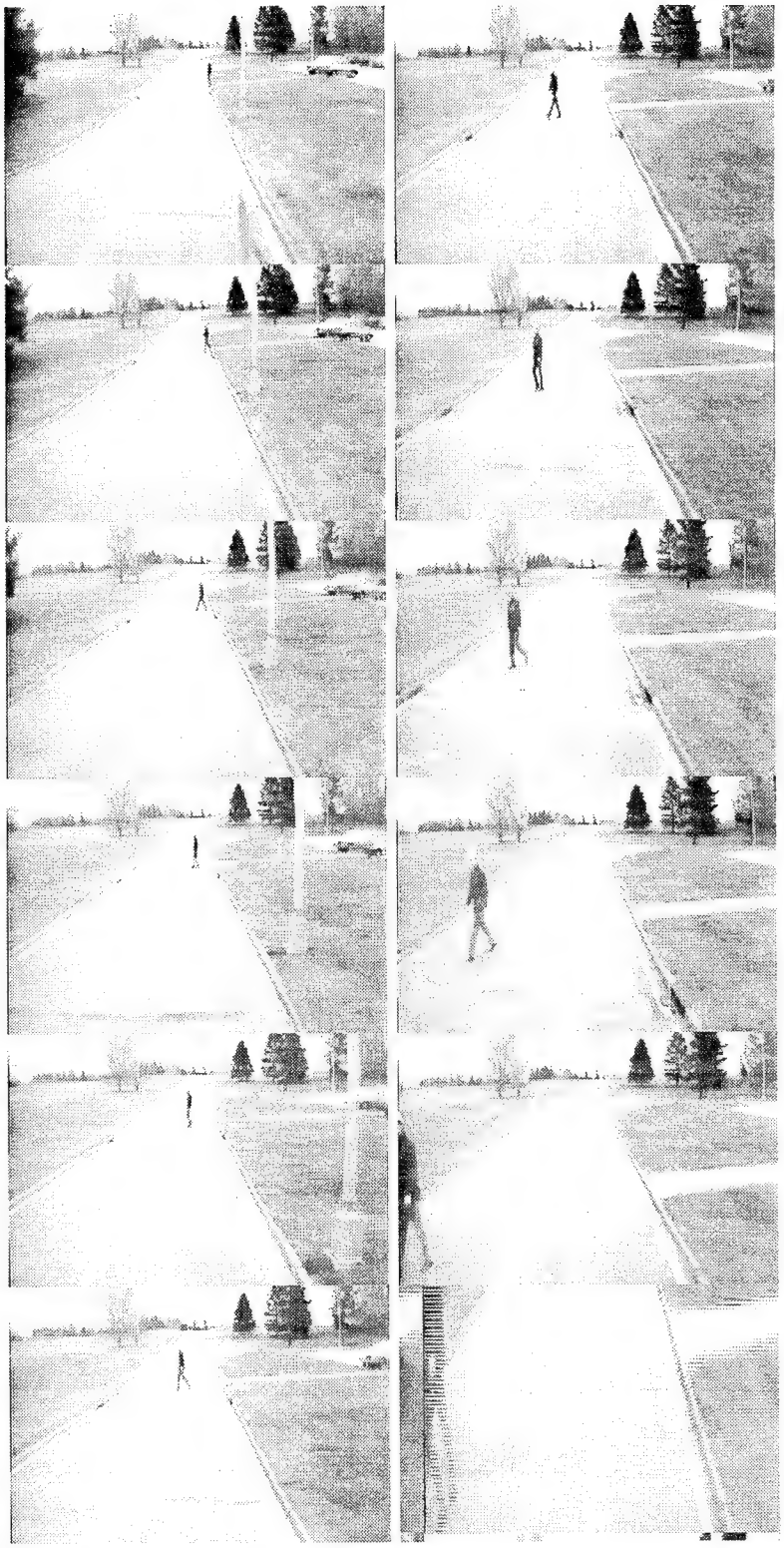

Figure 9: Frames from sequence of person walking across a road, taken by a camera mounted on a van moving at about 30 mph.

[htbp]

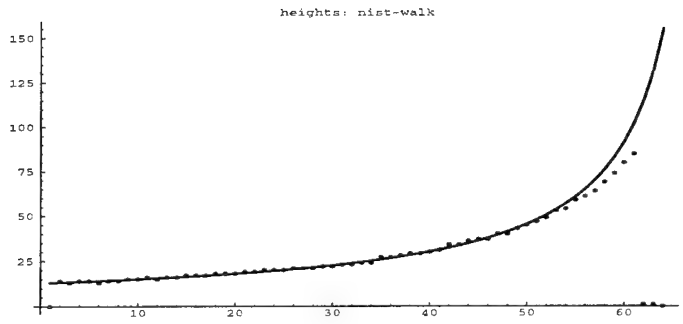

Figure 10: Actual image heights of the person (dotted), and the fitted model (solid)

database (which is fixed before hand). Unfortunately, the relation $h_t = h_0/(1 + w * t)$ is not linear in w and so we can not directly use the least squares technique to estimate w. Instead, we use an approximation to the model $h_t = h_0(1 - w * t)$ which is a good approximation if the term $w * t$ is small. We keep $w * t$ small by using the model in small temporal neighborhoods, so that t is very small (also $w = W/Z_0$ is small if the distance of the object from camera is large compared to the speed with which it is approaching or moving afar, which is true in most circumstances). Note that the relation $1/h_t = (1 + w * t)/h_0$ is linear in w, but using least squares to minimize the error between observed and model heights is not same as minimizing the error between observed and model inverted heights, and hence the estimate of w is not same, even though it may produce a reasonable estimate.

Thus the steps involved in detecting and compensating for changes in scale are: measure the image height of the object in each flow frame, use the heights in the last K frames to estimate the scale factor w and scale the image frame t to match the reference database scale, and recompute flow frame t.

To demonstrate the above technique, we have digitized an image sequence from the video recorded by NIST (National Institute of Standards & Technology) using a video camera mounted on a van looking straight ahead while the van is being driven on the road at about 30mph around the NIST grounds in Gaithersberg, Maryland. The image sequence is shown in Figure 9 which consists of a person walking across the street as the van is approaching. The image heights of the person for this sequence are hand-measured and plotted (dotted-line) in Figure 10. As can be seen, a linear fit over the entire image sequence is inappropriate for this data. We estimated the scale factor over the entire sequence using the least squares technique for inverted heights and plotted (solid-line)

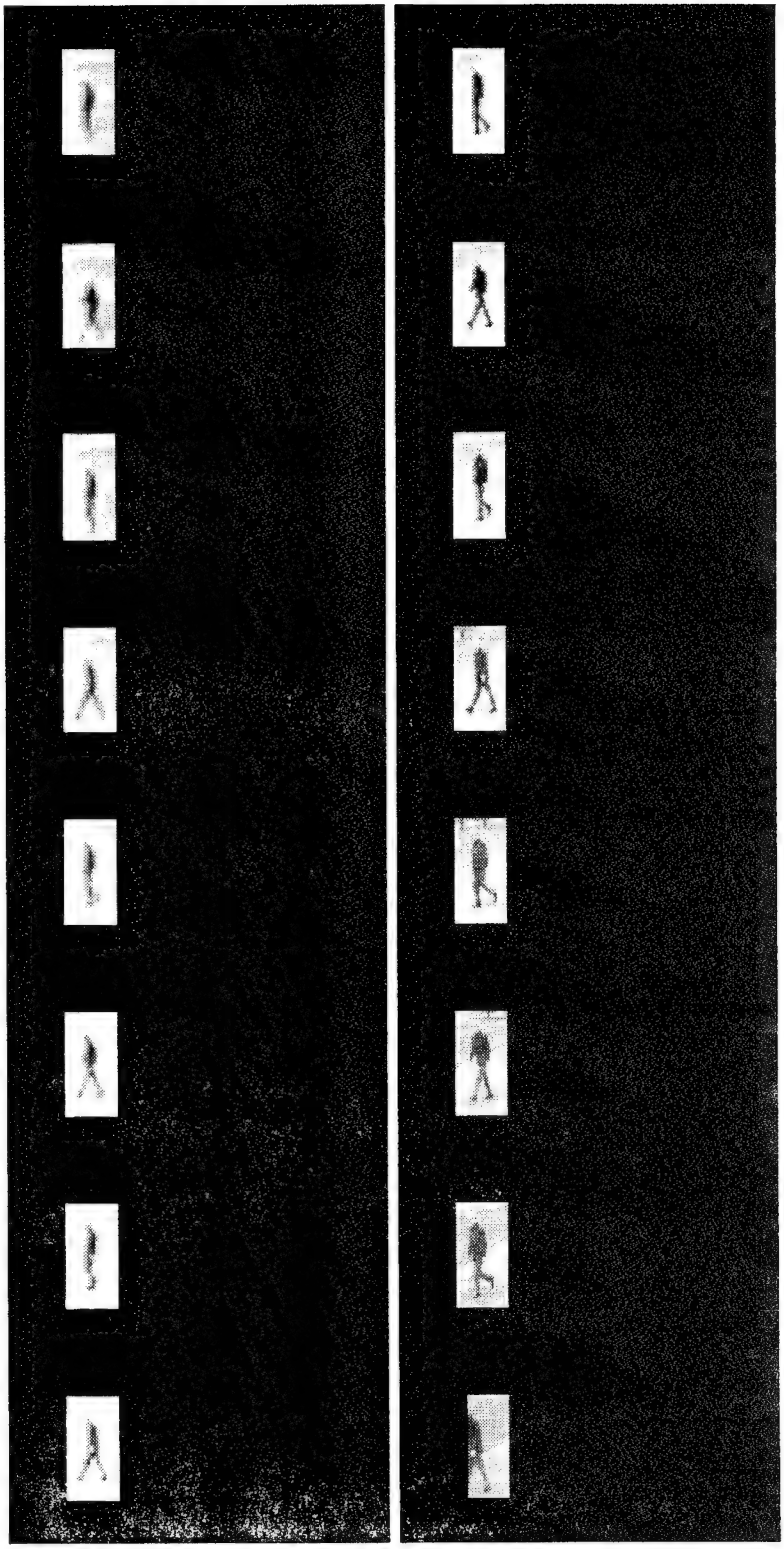

Figure 11: Frames from raod crossing sequence after tracking and compensating for scale changes.

the resulting fit to the data in the same figure. It gives a reasonably good approximation in the beginning where the distance from camera is large and at the right end a slight deviation from actual heights is seen where the distance of the person from camera is smaller compared to the speed of the vehicle. By using local linear models a better approximation is obtained and when the image frames are scaled and tracked as before, we obtain the stationary walking activity shown in Figure 11. The motion magnitude feature vector is computed for this image sequence and classification algorithm applied and it was correctly classified as walking (there being six other choices and a "don't know" condition).

5 Complete Algorithm

Given a gray-valued image sequence, the actor is detected, tracked and spatial scale changes are estimated. The image sequence is transformed so as to compensate for the spatial translation and scale changes of the actor. The resulting image sequence consists of the actor at the center of the image frame and at the same distance from the camera throughout the image sequence. The image frame is reduced to the size of the object and the motion is computed between successive image frames.

Each flow frame t is divided into a spatial grid of XxY cells and the motion magnitudes in each spatial cell are summed. Let $M(x, y, t)$ be the motion magnitude in flow frame t corresponding to spatial cell (x, y). According to the definition of a periodic activity, for each fixed (x, y), the signal $M(x, y, t)$ over time should be periodic. For each (x, y), we compute the periodicity index for each flow-signal and combine the individual periodicity indices to get a periodicity measure for the whole image frame. By thresholding the resulting periodicity measure it is possible to determine if the motion produced by the object is periodic. If it is found that there is sufficient periodicity in the motion, we proceed to compute the feature vector.

The frequency of the activity is found along with the periodicity measure and is used to divide the image sequence into a number of cycles. The flow frames are folded over temporally to obtain a single cycle of the activity, and the motion in different cycles is averaged to obtain a single combined activity cycle. The cycle is then divided into T temporal divisions and the motion magnitude is summed in each spatio-temporal cell. The resulting spatiotemporal motion template is used as a feature vector, which is compared to reference templates of known activities.

The classification method we have used is the nearest centroid algorithm, which is simple to implement and effectively shows the discriminating power of the feature vector. In this algorithm, feature vectors of a number of reference sample image sequences of each known activity are computed and their centroid is taken as the reference feature vector for the corresponding class. Feature vectors computed from a test sample image sequence are matched against the reference feature vectors of each class and the test sample is classified as the class corresponding to the centroid closest in euclidean distance.

To recognize an activity we need to set thresholds on the allowable distances between a test feature vector and the class centroids. This allows us to classify a motion as unknown if it is not sufficiently close to any of the known activities. A simple approach is to set the threshold of a reference class to be the average distance of class samples from the centroid. In this case, we would classify a test vector as belonging to class k if the test vector falls within a circular region around the centroid. A more accurate approach is to find the principal components of the reference vectors in each class and weigh the test vector elements inversely proportional to the corresponding coefficients

in the first principal component. (To avoid infinities produced by division by zero, we bound the maximum weight). The effect is that feature vector elements which are more consistent within the class are given higher priority in matching process, and the elements whose the variability is greater are given less weight. The net result of this procedure is to make the recognition regions around each class centroid ellipsoidal instead of circular. Note that this is not equivalent to determining a full covariance ellipsoid, which is theoretically optimal, but impractical in our case because of the high dimensionality of the feature vectors.

The feature vectors computed, while corrected for temporal scale (frequency) are not corrected for temporal translation (phase). For reference samples, we manually selected the phase so that the same time point of the activity is taken as the beginning of the cycle for all reference vectors considered. For test samples, we handle the unknown temporal phase, by simply trying a match at each possible phase and picking the best. Since the pattern matching phase of the algorithm currently represents only a small fraction of the total computational effort, the above method works efficiently compared to alternative methods of finding the temporal phase of the activity. The temporal resolution of the pattern is typically small (i.e., less than 10 samples per cycle),

The following is a step-by-step description of the periodic activity recognition algorithm:

- *Precompute centroids of classes:* Select K samples of each known class of activity and apply the following algorithm to obtain the reference feature vectors. For each class compute the centroids of the class by averaging the corresponding feature vectors. Find the first principal component within each class and normalize it so that the sum of its elements is one. These coefficients are used (inverted) in the matching process, as described above, to weight the elements of feature vectors.

- *Input:* The input to the algorithm is a digitized 256-level gray-valued image sequence consisting of at least four cycles of a periodic activity and a set of pixels corresponding to the object of interest in each image frame.

- *Output:* A known class into which the activity is classified by the algorithm.

- *Step 1.* Compute normal flow magnitude at each pixel between each successive pair of frames using the differential method.

- *Step 2.* Estimate scale changes and object motion locally, and compensate for both of them simultaneously. The result will be an image sequence consisting of only the object of interest which is centered in the frame. Recompute normal flow between successive frames of the compensated image sequence, an eliminate pixels whose flow is consistent with the known motion between frames.

- *Step 3.* For every flow frame of the sequence, divide the spatial activity window into a grid of size XxY and sum the motion magnitude in each cell. Compute the periodicity measure for each cell using the change in the motion magnitude signal over time. Compute the frequency histogram and select the dominant frequency and the associated periodicity measure. If the periodicity measure does not exceed a fixed threshold P, declare that the image sequence does not consist of an activity and stop. Otherwise obtain the frequency of the activity, filter out all motion that is not consistent with this period, and proceed to Step 4.

- *Step 4.* Average flow field over multiple cycles of the activity into one cycle. This gives a temporal window consisting of one cycle of flow field. Divide the spatiotemporal window of activity into three dimensional grid of XxYxT cells and sum the motion magnitudes within each cell. Divide the motion magnitudes by the size of the cell and overall average motion magnitude.

- *Step 5.* Match the test vector against the reference centroids while varying the temporal phase (i.e., compute T distances from the feature vector to each reference class centroid by shifting through T temporal divisions). Choose the minimum of these distances as the distance of the feature vector to the reference class. Select the class that gives the minimum distance centroid to the test feature vector. If the distance is less than the threshold previously determined for the class, output the classification; otherwise declare the presence of an unknown activity.

5.1 Real-Time Implementation

We have implemented the above algorithm on SGI architecture with four processors. The complexity is proportional to the number of pixels involved in the activity. Most of the computation time is spent on the normal flow computation. The implementation includes displaying the original image frame, gradient, flow, and the XxY grid motion magnitudes and the classification result at every frame of the image sequence. A sample screen through a run of the program is illustrated in figure 12. The total computation for each frame takes 60-80 milliseconds. Of course, more processors can be used for faster running times.

6 Conclusions

We have studied a class of motion patterns namely activities, and proposed efficient robust algorithm for their recognition using low-level statistical features based on motion. The algorithm utilizes a spatiotemporal motion magnitude template as a feature vector to classify the activity into one of several known classes. We have demonstrated the recognition of several activities including human walking which has been studied previously using model-based approaches. We have also shown that periodicity is an inherent low-level motion cue that can be exploited for robust detection of periodic phenomenon without prior structural knowledge. We illustrated the technique using real-world data, and showed that it robustly detects complex periodic activities, while excluding non-periodic motion. The recognition algorithms were implemented in near real-time using real-world image sequence samples.

The motion information is obtained by computing the normal flow between successive image frames using a simple differential technique. The issue of the accuracy of the motion computation is not of significance, as it is sufficient to show that useful features can be extracted from sufficiently reliable estimates of motion. The use of accurate motion estimation algorithms would only improve the reliability of the techniques described.

The affect of changing viewing angle in activity recognition has not been addressed in the current implementation. It was shown that a change in viewing angle of up to 30 degrees can be tolerated in the recognition of walking. However, motion templates derived from views having wider angular separation (compare walking towards the camera with walking perpendicular to the viewing axis), would exhibit substantial differences. A motion template consisting of three-dimensional motions

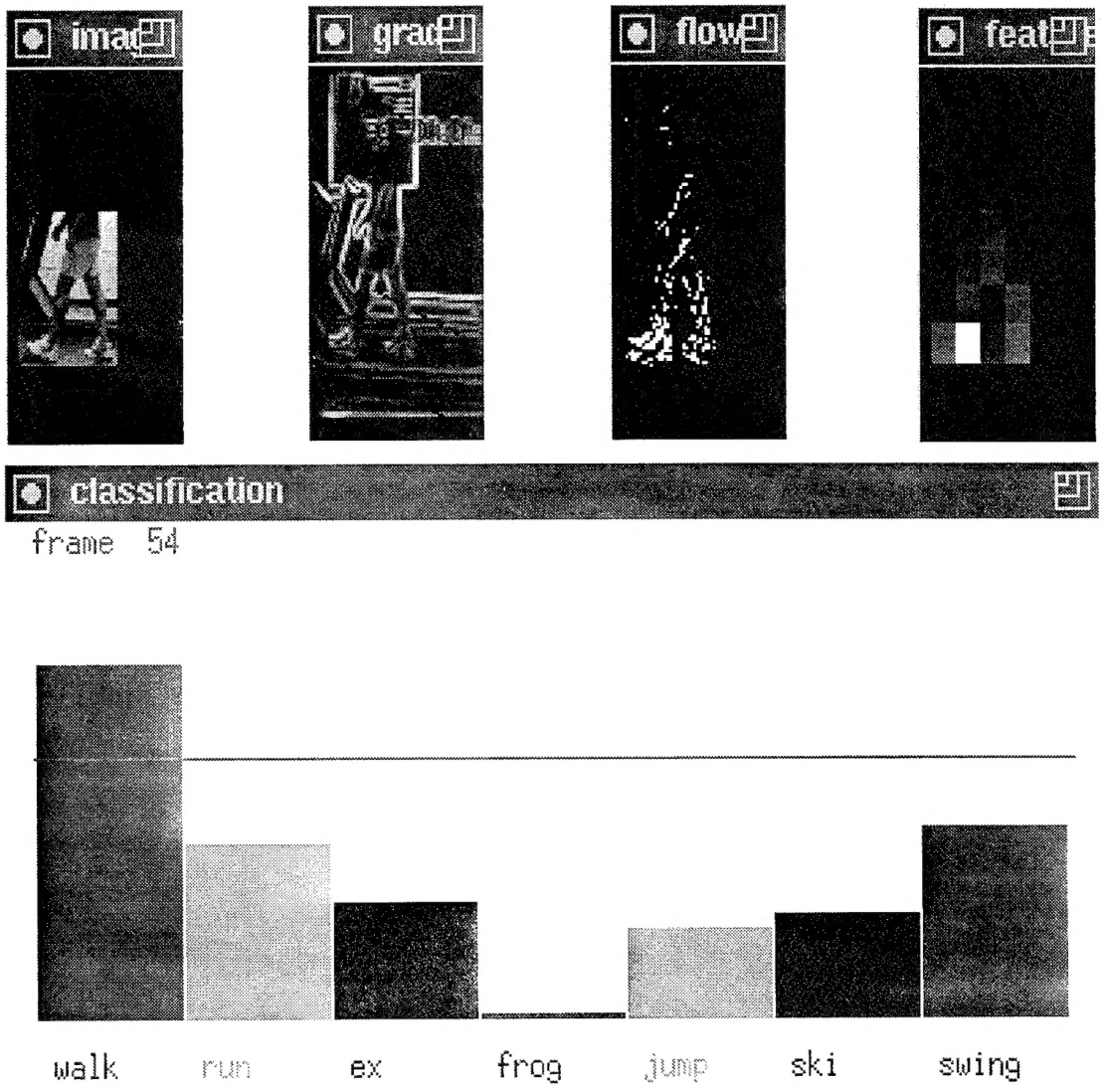

Figure 12: Real-time implementation screen: classification is performed at every frame

might be used to handle this, but this requires computation of three-dimensional motion vectors which is, as yet, an unreliable and computationally expensive process. The proposed method of handling different viewing angles is to incorporate multiple reference classes corresponding to different views of the same activity. It remains to be seen whether the additional classes can be added without adversely affecting the recognition rate.

6.1 Lip-reading

The feature vectors used in activity recognition proved useful in the case of lip-reading as well. Virginia de Sa [7] digitized image sequences of the mouth area during various utterances, and used the motion vectors computed by the method described here to classify the utterance. This information was used in conjunction with acoustic signals to improve the recognition rate of utterances. Visual data was collected for four persons (all male, one with moustache) for four utterances: /ba/, /wa/, /da/ and /ga/. The region imaged was centered around the mouth, covering the extent of the mouth horizontally, and extending vertically from the tip of the nose to the chin. A sample image sequence of an utterance is illustrated in 13. Ten frames of motion information around the utterance are used to compute a 5x5x5 spatiotemporal motion template of the utterance. Using a modified version of Kohonen's learning vector quantization 2.1 (LVQ2.1) [8], [19], with 60 code book vectors, the recognition rate of reference vectors was 85 percent and the recognition rate of test vectors was 69 percent.

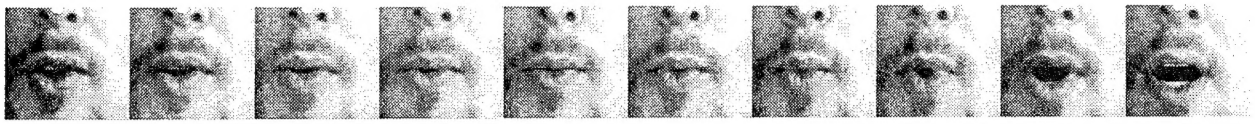

Figure 13: Sample image sequence of utterance /wa/.

6.2 Gesture Recognition

The use of spatiotemporal motion template to recognize activities can be extended to the case of gesture recognition as well. The main difficulty in gesture recognition is to isolate the temporal duration of the gesture. Once a reliable mechanism for identifying the start and end frames in any given image sequence is provided, the same method used in the recognition of activities can be used to compute a spatiotemporal motion template and classify the gesture.

References

[1] M. Allmen and C.R. Dyer. Cyclic motion detection using spatiotemporal surface and curves. In *Proc. Int. Conf. on Pattern Recognition*, pages 365–370, 1990.

[2] C. H. Anderson, P. J. Burt, and G. S. van der Wal. Change detection and tracking using pyramid transform techniques. In *Proc. SPIE Conference on Intelligent Robots and Computer Vision*, pages 300–305, 1985.

[3] N.I. Badler. *Temporal Scene Analysis: Conceptual Descriptions of Object Movements.* PhD thesis, Univ of Toronto, 1975.

[4] H.W. Chun. A representation for temporal sequence and duration in massively parallel networks: Exploiting link connections. In *Proc. AAAI*, 1986.

[5] J. Crane. Imaginal behaviour of trinidad butterfly. *Zoologica*, 40:167–196, 1955.

[6] J.E. Cutting. Six tenets for event perception. *Cognition*, pages 71–78, 1981.

[7] Virginia R. de Sa. *Unsupervised Classification Learning from Cross-Modality Structure in the Environment.* PhD thesis, Computer Science Department, Univ of Rochester, 1994.

[8] Virginia R. de Sa and Dana H. Ballard. Self-teaching through correlated input. In *Computation and Neural Systems 1992*, pages 437—441. Kluwer Academic, 1993.

[9] J.E. Elman. Finding structure in time. Technical Report 8801, Center for Research in Language, Univ. of California, San Diego, 1988.

[10] J.P. Ewart. Neuroethology of releasing mechanisms: Prey-catching in toads. *Behavioral and Brian Sciences*, 10:337–405, 1987.

[11] J.E. Feldman. Time, space and form in vision. Technical Report 244, University of Rochester, Computer Science Department, 1988.

[12] K.E. Finn and A.A. Montgomery. Automatic optically-based recognition of speech. *Pattern Recognition Letters*, 8:159–164, 1988.

[13] K. Gould, K. Rangarajan, and M.A. Shah. Detection and representation of events in motion trajectories. In Gonzalez and Mahdavieh, editors, *Advances in Image Processing and Analysis.* SPIE Optical Engineering Press, 1992.

[14] K. Gould and M. Shah. The trajectory primal sketch: A multi-scale scheme for representing motion characterestics. In *IEEE Conf. Computer Vision and Pattern Recognition*, pages 79–85, 1989.

[15] E.C. Hildreth and C. Koch. The analysis of visual motion from computational theory to neural mechanisms. *Annual Review of Neuroscience*, 1987.

[16] D.D. Hoffman and B.E. Flinchbuagh. The interpretation of biological motion. *Biological Cybernatics*, pages 195–204, 1982.

[17] G. Johansson. Visual perception of biological motion and a model for its analysis. *Perception and Psychophysics*, 14:201–211, 1973.

[18] B.H. Juang and L.R. Rabiner. Mixture autoregressive hidden markov models for speech signals. *IEEE Trans. Acoustics, Speech and Signal Processing*, 6:1404–1413, 1985.

[19] Teuvo Kohonen. Improved versions of learning vector quantization. In *IJCNN International Joint Conference on Neural Networks*, volume 1, pages I–545–I–550, 1990.

[20] D. Koller, N. Heinze, and H.-H. Nagel. Algorithmic characterization of vehicle trajectories from image sequences of motion verbs. In *Proc. of IEEE Computer Vision and Pattern Recognition*, pages 90–95, 1991.

[21] R.C. Nelson. Qualitative detection of motion by a moving observer. In *Proc. of IEEE CVPR*, pages 173–178, 1991.

[22] J. O'Rourke and N.I. Badler. Model-based image analysis of human motion using constraint propagation. *PAMI*, 3(4):522–537, 1980.

[23] A. Pentland and K. Mase. Lip reading: Automatic visual recognition of spoken words. Technical Report 117, M.I.T. Media Lab Vision Science, 1989.

[24] E.D. Petajan, B. Bischoff, and N.M. Brooke. An improved automatic lipreading system to enhance speech recognition. In *SIGCHI'88: Human Factors in Computing Systems*, pages 19–25, 1988.

[25] R. Polana and R.C. Nelson. Temporal texture recognition. In *Proc. of CVPR*, pages 129–134, 1992.

[26] R. Polana and R.C. Nelson. Detecting activities. *Journal of Visual Communication and Image Representation*, 5(2):172–180, 1994.

[27] R.F. Rashid. *LIGHTS: A System for Interpretation of Moving Light Displays.* PhD thesis, Computer Science Dept, University of Rochester, 1980.

[28] J.R. Rhyne and C.G. Wolf. Gestural interfaces for information processing applications. Technical Report 12179, IBM Research Report, 1986.

[29] S.M. Seitz and C.R. Dyer. Affine invariant detection of periodic motion. In *Proceedings of CVPR*, 1994.

[30] R.H. Smythe. *Vision in the Animal World.* St. Martin's Press, NY, 1975.

[31] D. W. Tank and J. J. Hopfield. Concentrating information in time: analog neural networks with applications to speech recognition problems. In *Proceedings of the First International Conference on Neural Networks*, pages 455–468, 1987.

[32] N. Tinbergen. *The Study of Instinct.* Oxford: Clarendon Press, 1951.

[33] R. Y. Tsai and T. S. Huang. Estimating 3-d motion parameters of a rigid planar patch i. *IEEE ASSP*, 30:525–534, 1981.

[34] K. 'Frisch von'. *Bees: Their Vision, Taste, Smell and Language.* Moscow,IL, 1955.

[35] E. Wolf and G. Zerrahn-Wolf. Flicker and the reactions of bees to flowers. *Journal of Gen. Physiol.*, 20:511–518, 1936.